

REMOTE STORAGE DISK CONTROL DEVICE WITH FUNCTION TO
TRANSFER COMMANDS TO REMOTE STORAGE DEVICES

CROSS-REFERENCE TO RELATED APPLICATIONS:

5 The present invention is related to Japanese patent application No. 2003-400513 filed in Japan on November 28, 2003, and Japanese patent application No. 2003-325082 filed in Japan on September 17, 2003, which are incorporated herein by reference.

10 BACKGROUND OF THE INVENTION:

1. Field of the Invention

 The present invention relates to a method for controlling a storage device system, a storage device system, and a storage device.

15 2. Related Background Art

 Disaster recovery in information processing systems is attracting attention. As a technology to realize such disaster recovery, a technology in which a copy of data stored in a storage device that is installed in a primary site is also managed by a storage device that is installed in a remote site located away from the primary site is known. By using the data stored in the storage device installed at the remote site when the primary site is hit by a disaster, processings that are performed at the primary site can be continued at the remote site.

 For data transfer from the primary site to the remote site, a method

in which data is exchanged between an information processing device at the primary site and an information processing device at the remote site is known. The information processing device at the primary site transfers a copy of data that is written in the storage device at the primary site to the information processing device at the remote site. The information processing device at the remote site that has received the copy of data sends a request to write the data in the storage device at the remote site.

When data is stored as a backup by the method described above, a substantially large amount of data flow occurs on the network between the information processing devices. This causes a variety of problems such as an increased interface processing load on the information processing devices, delays in other data transmissions to be conducted between the information processing devices, and the like. Also, the method described above needs software to control data backup to be installed in each of the information processing devices. For this reason, management works such as upgrading the software and the like need to be performed on all of the information processing devices that execute data backup processings, which increases the management cost.

SUMMARY OF THE INVENTION:

The present invention has been made in view of the problems described above, and relates to a storage device system, a storage device and a method for controlling a storage device system.

In accordance with an embodiment of the present invention, there is

provided a method for controlling a storage device system that includes: at least one information processing device, a first storage device equipped with a first storage volume, and a second storage device equipped with a second storage volume, wherein the information processing device and the first storage device are communicatively connected to one another, the first storage device and the second storage device are communicatively connected to one another, the information processing device is equipped with a first write request section that requests to write data in the first storage device according to a first communications protocol, and the first storage device is equipped with a second write request section that requests to write data in the second storage device according to a second communications protocol. The method comprises: a step in which the information processing device sets a first instruction to be executed at the second storage device as first data; a step in which the information processing device sends a request to write the first data in the first storage volume to the first write request section; a step in which, when the first data written in the first storage volume is an instruction to the second storage device, the first storage device sends a request to write the first data in the second storage volume to the second write request section; and a step in which the second storage device executes the first instruction that is set as the first data written in the second storage volume.

It is noted that the information processing device may be, for example, a personal computer, a work station or a mainframe computer. The storage device may be, for example, a disk array device or a

semiconductor storage device. The storage volume may be a storage resource that includes a physical volume that is a physical storage region provided by a disk drive, and a logical volume that is a storage region logically set on the physical volume. Also, the communications protocol
5 may be, for example, a WRITE command stipulated by a SCSI (Small Computer System Interface) standard. As a result, without adding new commands to the operating system, the information processing device can make the second storage device to execute the first command.

Here, for example, when the first command is a command to read
10 data of the first storage device, the second storage device can have a copy of the data of the first storage device according to an instruction from the information processing device. Therefore the present method can reduce the amount of data communicated between the information processing devices in the data backup management. Also, software for controlling
15 data backup does not have to be installed on all of the information processing devices that are performing data backup, which lowers the management costs.

Other features and advantages of the invention will be apparent from the following detailed description, taken in conjunction with the
20 accompanying drawings that illustrate, by way of example, various features of embodiments of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS:

FIG. 1 schematically shows a system configuration of an

information processing system in accordance with an embodiment of the present invention.

FIG. 2 schematically shows a structure of a disk array device in accordance with an embodiment of the present invention.

5 FIG. 3 shows a LUN map information table in accordance with an embodiment of the present invention.

FIG. 4 schematically shows a diagram illustrating a data writing operation using virtual volumes in accordance with an embodiment of the present invention.

10 FIG. 5 shows a command device management table in accordance with an embodiment of the present invention.

FIG. 6 shows a command device interface in accordance with an embodiment of the present invention.

15 FIG. 7 is a schematic diagram illustrating execution of commands set at command devices in accordance with an embodiment of the present invention.

FIG. 8 shows a flowchart of an operation to control a command device in an information processing device in accordance with an embodiment of the present invention.

20 FIG. 9 shows a flowchart of an operation to control a command device in a storage device in accordance with an embodiment of the present invention.

FIG. 10 shows a pair management table in accordance with an embodiment of the present invention.

FIG. 11 shows a schematic diagram illustrating a pair forming processing in accordance with an embodiment of the present invention.

FIG. 12 shows a relation between a primary volume and a primary journal in accordance with an embodiment of the present invention.

5 FIG. 13 is a schematic diagram illustrating a journal acquisition processing in accordance with an embodiment of the present invention.

FIG. 14 is a schematic diagram illustrating a restoration processing in accordance with an embodiment of the present invention.

FIG. 15 shows journal data regions of a primary journal and an
10 auxiliary journal in accordance with an embodiment of the present invention.

FIG. 16 shows a flowchart of a journal acquisition processing and a restore processing performed in information processing devices in accordance with an embodiment of the present invention.

15 FIG. 17 is a schematic diagram illustrating a swap processing in accordance with an embodiment of the present invention.

FIG. 18 is a schematic diagram illustrating a state in which the swap processing of the present embodiment is completed.

FIG. 19 shows a flowchart of a swap processing in a second
20 information processing device in accordance with an embodiment of the present invention.

FIG. 20 shows a flowchart of a swap processing in a first storage device in accordance with an embodiment of the present invention.

FIG. 21 shows a flow chart of a swap processing in a second storage

device in accordance with an embodiment of the present invention.

FIG. 22 schematically shows a diagram of an information processing system in accordance with an embodiment of the present invention in which a command is sent to a third storage device.

5 FIG. 23 schematically shows a diagram of an information processing system in accordance with an embodiment of the present invention in which a command is sent to a second storage device and a third storage device.

FIG. 24 shows a synchronous pair management table in accordance
10 with an embodiment of the present invention.

FIG. 25 shows a path information management table in accordance with an embodiment of the present invention.

FIG. 26 shows a command interface in accordance with an embodiment of the present invention.

15 FIG. 27 schematically shows a diagram illustrating a storage device system in which an information processing device sends a command to a second storage device, using a command interface in accordance with an embodiment of the present invention.

FIG. 28 schematically shows a diagram illustrating a storage device
20 system in which an information processing device sends a command to a third storage device, using a command interface in accordance with an embodiment of the present invention.

FIG. 29 shows an example of a path information management table that sets a third storage device in accordance with an embodiment of the

present invention.

FIG. 30 shows a flowchart of processings in which an information processing device sends a command to a storage device, using a command interface in accordance with an embodiment of the present invention.

5 FIG. 31 shows a flowchart of processings in which a storage device sends or executes a command, using a command interface in accordance with an embodiment of the present invention.

FIG. 32 shows a flowchart of processings in which a storage device obtains edited data of a command executed, using a command interface in
10 accordance with an embodiment of the present invention.

FIG. 33 shows a flowchart of processings in which a storage device transfers or executes a command without deleting a transfer destination address of a command interface in accordance with an embodiment of the present invention.

15 FIG. 34 shows a flowchart of processings in which a storage device obtains edited data of a command executed without deleting a transfer destination address of a command interface in accordance with an embodiment of the present invention.

FIG. 35 schematically shows a diagram illustrating a storage device
20 system in which the shortest transfer path is searched for sending a command interface in accordance with an embodiment of the present invention.

FIG. 36 shows an example of a connection information management table in accordance with an embodiment of the present invention.

FIG. 37 shows a flowchart of processings in which the shortest transfer path among storage devices is searched for transferring a command interface or executing a command in accordance with an embodiment of the present invention.

5 FIG. 38 shows an example of a command interface used when having a plurality of storage devices execute commands in accordance with an embodiment of the present invention.

FIG. 39 schematically shows a diagram illustrating a storage device system in which an information processing device sends commands to a
10 second storage device and a third storage device, using a command interface in accordance with an embodiment of the present invention.

FIG. 40 shows a flowchart of processings in which an information processing device sends commands to a plurality of storage devices, using a command interface in accordance with an embodiment of the present
15 invention.

FIG. 41 shows a flowchart of processings performed by a storage device system in accordance with an embodiment of the present invention in which a command interface is transferred among a plurality of storage devices or commands are executed, using the command interface.

20

PREFERRED EMBODIMENTS OF THE INVENTION:

[Example of Overall Structure]

FIG. 1 schematically shows an overall structure of an information processing system including a storage device system in accordance with an

embodiment of the present invention. The information processing system of the present embodiment includes an information processing device 11, and at least a first storage device 10 and a second storage device 20. The first storage device 10 is equipped with a logical volume(s) 30 on which the first storage device performs data input/output processings (hereafter referred to as a “first logical volume(s)”), and the second storage device 20 is equipped with a logical volume(s) 40 on which the second storage device performs data input/output processings (hereafter referred to as a “second logical volume(s)”).

The information processing device 11 and the first storage device 10 are communicatively connected to each other via a first network 50. The first network 50 may be, for example, a LAN (Local Area Network), a SAN (Storage Area Network), an iSCSI (Internet Small Computer System Interface), an ESCON (Enterprise Systems Connection) ®, or a FICON (Fibre Connection) ®.

The first storage device 10 and the second storage device 20 are communicatively connected to each other via a second network 60. The second network 60 may be, for example, a Gigabit Ether Net ®, an ATM (Asynchronous Transfer Mode), or a public telephone line.

[Information Processing Device]

The information processing device 11 may be a computer that is equipped with a CPU (Central Processing Unit), memories, and other devices. The information processing device 11 may be a personal

computer, a work station or a mainframe computer. The information processing device 11 may be composed of a plurality of computers that are mutually connected. An operating system is operating on the information processing device 11, and application software is operating on the
5 operating system.

[Storage Device]

FIG. 2 shows a structure of a disk array device, which is described as an example of the first storage device 10 and the second and 20.
10 Instead of the disk array device, the first and second storage devices 10 and 20 may be any appropriate devices, such as, for example, semiconductor storage devices. For example, the disk array device 10 is equipped with various components including a channel control section 201, a remote communications interface 202, disk control sections 203, a shared
15 memory 204, a cache memory 205, a switching control section 206 that is composed of cross bus switches that communicatively connect the components described above, a management terminal 207, and memory devices 208. The first and second storage devices 10 and 20 may have the same structure.

20 The cache memory 205 is used to temporarily store data that is exchanged mainly between the channel control section 201 and the disk control sections 203. For example, when a data input/output command which the channel control section 201 receives from the information processing device 11 is a write command, the channel control section 201

writes in the cache memory 205 write data received from the information processing device 11. Also, an appropriate one of the disk control devices 203 reads the data written in the cache memory 205, and writes the same in the memory devices 208.

5 The disk control section 203 reads a data I/O request stored in the shared memory 204 written by the channel control section 201, and executes data writing processing or data reading processing with respect to the memory devices 208 according to a command set at the data I/O request (for example, a command according to a SCSI standard). The disk
10 control section 203 writes in the cache memory 205 data that has been read out from the memory devices 208. Also, the disk control section 203 transmits to the channel control section 201 notifications, such as, for example, a data write completion notification and a data read completion notification. The disk control section 203 may be equipped with a function
15 to control the memory devices 208 with RAID levels (for example, 0, 1, 5) stipulated in the so-called RAID (Redundant Array of Inexpensive Disks) method.

 The memory devices 208 may be, for example, hard disk devices. The memory devices 208 may be provided in one piece with or separately
20 as independent devices from the disk array device. Storage regions provided by the memory devices 208 at each site are managed in units of logical volumes 209, which are volumes that are logically set on the storage regions. Data can be written in or read from the memory devices 208 by designating LUNs (Logical Unit Numbers) that are identifiers

appended to the corresponding logical volumes 209. Also, the logical volumes 209 are managed in units of a predetermined data amount such as units of 512 Kb, such that input and output of data in this predetermined unit are conducted. Each of the units is called a logical block, and each of the logical blocks is appended with a logical block address (hereafter referred to as a "LBA") that indicates positional information of the logical block.

The management terminal 207 may be a computer for maintaining and managing the disk array device and the memory devices 208.

Changes in the software and parameters to be executed by the channel control section 201 and the disk control section 203 can be conducted by giving instructions from the management terminal 207. The management terminal 207 can be in a form that is built in the disk array device, or can be provided independently from the disk array device.

The remote communications interface 202 is a communications interface (i.e., a channel extender) that is used for data transfer to another storage device. A copy of data is transferred in a remote copy operation to be described below through this remote communications interface 202. The remote communications interface 202 converts the interface of the channel control section 201 (for example, an interface such as an ESCON ® interface or a FICON ® interface) to a communications method of the second network 60, whereby data transfer with the other storage device can be realized.

Besides the structure described above, the disk array device may

have a structure that functions as a NAS (Network Attached Storage) configured to accept data input/output requests through designating file names from the information processing device 11 according to a relevant protocol such as a NFS (Network File System).

5 The shared memory 204 can be accessed from both of the channel control section 201 and the disk control section 203. The shared memory 204 is used for delivering data input/output request commands, as well as for storing management information for the storage devices 10 and 20, and the memory devices 208. In the present embodiment, the shared memory
10 204 stores a LUN map information table 301 shown in FIG. 3, a command device management table 501 shown in FIG. 5, and a pair management table 1001 shown in FIG. 10.

[Virtual Volume]

15 As described above, the logical volumes 209 are storage regions that are logically set on the physical volumes. Also, by using “virtual volumes” as logical volumes, the storage devices 10 and 20 on which the logical volumes 209 are set can be differentiated from other storage devices that are equipped with physical volumes correlated with the logical volumes
20 209.

To realize this function, the first storage device 10 stores a LUN map information table 301 shown in FIG. 3. The LUN map information table 301 describes information relating to the logical volumes 209 that are handled by the first storage device 10. For example, in the present

embodiment, the LUN map information table 301 includes entries of “LUN,” “Target” and “Mapping LUN.”

Each entry at “LUN” describes a LUN for each of the logical volumes. When a logical volume 209 is a virtual volume, a storage device
5 that is equipped with the logical volume 209 correlated with the virtual volume is set at “Target.” Furthermore, a LUN of the logical volume 209 correlated with the virtual volume is set at “Mapping LUN.” In other words, when there is a description at “Mapping LUN,” it means that the corresponding logical volume is a virtual volume.

10 Details of the LUN map information table 301 may be registered, for example, by an operator through the management terminal 207 that is connected to the first storage device 10.

The first storage device 10 uses the LUN map information table 301 described above and provides the second logical volume 40 of the second
15 storage device 20 to the information processing device 11 by a mechanism to be described below as if the second logical volume 40 were the first logical volume 30 of the storage device 10. In other words, the information processing device 11 can make data input/output requests, which are to be issued to the logical volume 209 of the second storage device 20, to the first
20 storage device 10.

Processings by the storage device system, which take place when a data input/output request transmitted from the information processing device 11 is a data write request, will be described with reference to FIG. 4.

The information processing device 11 is equipped with a first write

request section 401 that writes data in the first storage device 10 according to a first communications protocol. Upon receiving a data write request from the first write request section 401 (S401), the first storage device 10 writes in the cache memory 205 data to be written that has been
5 received with the data write request.

A data transfer section 402 of the first storage device 10 refers to the LUN map information table 301, and confirms as to whether or not a mapping LUN is set for a first logical volume 30 that is set in the data write request. If a second logical volume 40 is set as the mapping LUN,
10 the data transfer section 402 transfers to a second write request section 403 a request to write the data in the second logical volume 40 according to a second communications protocol. In this embodiment, the second write request section 403 makes data write requests to the second storage device 20 according to the second communications protocol. The second
15 storage device 20 receives the data write request from the second write request section 403, and writes the data in the second logical volume 40 (S402).

It is noted that the first communications protocol and the second communications protocol are for example WRITE commands stipulated by
20 a SCSI standard. Accordingly, the data write interfaces at the first storage device 10 and the second storage device 20 do not need to be changed.

The write processing has been so far described. It is noted however that a read processing to read data from a logical volume is also performed

in a manner similar to the write processing except that data is transferred in an opposite direction with respect to the data transfer direction in the write processing.

As describe above, in the storage device system in accordance with
5 the present embodiment, the information processing device 11 accesses the second logical volume as if the second logical volume were a logical volume on the first storage device 10.

[Command Device]

10 Each of the storage devices 10 and 20 is equipped with a “command device” for controlling special commands. The command device is used to convey commands from the information processing device 11 to the storage devices 10 and 20, and the storage devices 10 and 20 can execute commands that are stored in the command devices. What makes the
15 special commands different from ordinary commands is that the command devices are the logical volumes 209. Functions of the command device will be described below.

FIG. 5 shows a command device management table 501 that is stored in each of the storage devices 10 and 20. The command device
20 management table 501 contains entries such as “Device” and “Command Device LUN.” Entries at “Device” indicate as to which of the storage devices 10 and 20 correspond to which of the command devices. Each entry at “Command Device LUN” sets a LUN of each logical volume 209 which expresses the entity of the corresponding command device.

Details of the command device management table 501 may be registered, for example, by an operator through the management terminal 207 that is connected to each of the storage devices 10 and 20.

The command device management table 501 of each of the storage devices 10 and 20 can register command devices of other storage devices (that may be similar to the storage device 10 or 20). When the command devices of the other storage devices are registered, LUNs of virtual volumes, which correspond to the LUNs of the command devices of the other storage devices are registered at the entries “Command Device LUN.”

FIG. 6 shows an example of a command device interface 601, which is a data format of data that is written in a command device. The command device interface 601 is composed of a control parameter, an input parameter, and edited data. The control parameter is composed of a “process number” that indicates a command to be executed by a relevant storage device, and a “presence or absence of edited data” that indicates whether or not data is outputted as a result of the execution of the command. The input parameter sets parameter information that is used when executing the command. Also, the edited data sets data that is outputted as a result of executing the command.

An outline of a process flow to execute a command using a command device will be described with reference to FIG. 7. The information processing device 11 is equipped with a command setting section 701 and a command transmission section 702. The command setting section 701

generates data that sets in a command interface 601 a “process number” of a command to be executed by the first storage device 10 and its “presence or absence of edited data.” The command transmission section 702 transmits to the first write request section 401 a request to write the data in a first logical volume, which is a command device of the first storage device 10, according to a first communications protocol.

The first storage device 10 is equipped with a command execution section 703. The command execution section 703 is equipped with a pair management section 704, a copy forming section 705, a restore section 706, a journal storing section 707, a journal acquisition section 708 and a journal stop section 709, which control pairs of the logical volumes 209 to be described below.

The command execution section 703 refers to a command device management table 501, and obtains a LUN of a command device that corresponds to the first storage device 10 (S701). The command execution section 703 refers to the command device (S702) and, if data in the form of the command device interface 601 exists, executes a command designated by a process number indicated in the data.

Referring to flow charts in FIGS. 8 and 9, processes executed by the information processing device 11 and the storage devices 10 and 20 will be described. First, the information processing device 11 sets a “process number” and a “presence or absence of edited data” in first data in the form of the command device interface 601 (S801). Then, the information processing device 11 refers to a command device management table 501

stored in the storage device 10, and obtains a LUN of a relevant command device of a storage device which executes the command. In order to write the created first data at the LUN obtained, the information processing device 11 transmits to the storage device 10 a write request designating
5 the LUN (S802).

Upon receiving the write request, the storage device 10 writes the first data in the command device at the designated LUN.

It is noted that command devices are logical devices that are defined on storage areas of a plurality of storage devices, like the logical volumes
10 209, and write requests to the command devices are transmitted based on the same communications protocol as that for write requests transmitted to the logical volumes 209.

The storage device 10 refers to the command device management table 501, to specify LUNs of command devices that the storage device 10
15 itself should refer to, and monitors whether or not the command devices have data written therein (S901). When the first data is found written in any of the command devices under observation, the storage device 10 executes the command designated by the process number in the first data (S902). Having completed the execution of the command, the storage
20 device 10 confirms whether edited data of the first data is present or absent (S903). When edited data is absent, the storage device 10 deletes the first data from the command device (S906). When edited data is present, the storage device 10 sets data outputted as a result of execution of the command as edited data (S904).

The information processing device 11 confirms whether edited data for the command is present or absent (S803); and transmits to the storage device 10 a read request to read the edited data of the first data when the edited data is present (S804). Upon receiving the edited data from the
5 storage device 10 (Yes at S805), the information processing device 11 completes the processing. It is noted that the read request is transmitted based on the same communications protocol for read requests for the logical volumes 209 other than the command device.

When the edited data exists, after receiving the read request for the
10 edited data from the information processing device 11 (S905), the storage device 10 deletes the first data from the command device (S906).

In this manner, read or write requests that are used by the information processing device 11 for reading or writing data from and to ordinary logical volumes of the storage device 10, the information
15 processing device 11 can transfer commands to the storage device 10.

Also, by using the virtual volumes, the information processing device 11 can transfer commands to the second storage device 20 through the first storage device 10, such that the second storage device 20 can execute the commands.

20 It is noted that, when the information processing device 11 requests the storage device 10 and 20 to execute a “pair formation,” “journal acquisition,” “acquisition of processing state of journal,” “restore” or “swap” processing to be described below, the information processing device 11 uses the virtual volumes and command devices.

[Pair Formation]

Next, a description will be made as to a method for storing a copy of data in the logical volume 209 of the first storage device 10 in the logical
 5 volume 209 of the second storage device 20.

FIG. 10 shows a pair management table 1001. In the pair management table 1001, a column of "COPY SOURCE DEVICE" indicates storage devices (10 or 20) to which those of the logical volumes of copy sources belong (hereafter referred to as "primary volumes"). A column of
 10 "COPY DESTIANTION DEVICE" indicates storage devices (10 or 20) to which those of the logical volumes of copy destinations belong (hereafter referred to as "auxiliary volumes"). Also, a column of "PRIMARY LUN" indicates LUNs of the corresponding primary volumes, and a column of "AUXILIARY LUN" indicates LUNs of the corresponding auxiliary
 15 volumes. Each correspondence between a primary volume and an auxiliary volume is called a "pair." Those of the logical volumes 209 for storing journals (to be described below) are assigned to the primary volumes and auxiliary volumes. A column of "PRIMARY JOURNAL LUN" sets LUNs of the logical volumes 209 of the journals assigned for the
 20 primary volumes (hereafter referred to as "primary journals"). A column of "AUXILIARY JOURNAL LUN" sets LUNs of the logical volumes 209 of the journals assigned for the auxiliary volumes (hereafter referred to as "auxiliary journals").

Any one of appropriate methods for assigning the logical volumes

209 for storing the journals can be used. For example, the user himself/herself may designate those of the logical volumes 209 to be used as the journals, or the information processing device 11 may select appropriate unused ones of the logical volumes 209.

5 Referring to FIG. 11, an example of a process flow in forming pairs will be described. In this example, it is assumed that the first storage device 10 is equipped with a third logical volume and a fifth logical volume, and the second storage device 20 is equipped with a fourth logical volume and a sixth logical volume. The information processing device 11
10 transmits a command to the first storage device 10 and the second storage device 20 for forming a pair of the third logical volume as being a primary volume and the fourth logical volume as being an auxiliary volume, and a pair of the fifth logical volume as being a primary journal and the sixth logical volume as being an auxiliary journal (S1101, S1102). The pair
15 management sections 704 of the first and second storage devices 10 and 20 store information indicating the states of the pairs in the pair management tables 1001 of the respective storage devices 10 and 20. The copy forming section 705 of the second storage device 20 transmits to the first storage device 10 a read request to read data in the primary volume;
20 and upon receiving from the first storage device 10 a copy of the data in the primary volume, the second storage device 20 writes the data in the auxiliary volume (S1103). By this operation, the data in the primary volume and the data in the auxiliary volume can be matched with each other. A processing that brings the primary volume in conformity with the

auxiliary volume by a pair forming instruction is called a “initial copy” processing.

Also, the journal storage section 707 of the first storage device 10 starts a processing to obtain a copy of the data written in the primary volume and its positional information in the primary journal. The correlation between the primary volume and the primary journal is described hereunder with reference to FIG. 12. The primary journal is composed of a meta data region 1201 and a journal data region 1202. The journal storage section 707 of the first storage device 10 stores a copy of the data written in the primary volume (hereafter referred to as “journal data”) in the journal data region 1202. Also, the journal storage section 707 of the first storage device 10 stores in the meta data region 1201 the time when data 1203 is updated in the primary volume, LBA(s) 1204 of the data 1203, LBA(s) 1206 of the corresponding journal data region, and the data length of the updated data. Also, the auxiliary journal is composed of a meta data region 1201 and a journal data region 1202 like the primary journal.

Also, by using a method similar to the above, a copy of data in the logical volume 209 of the second storage device 20 can be stored in the logical volume 209 of the first storage device 10 by an instruction from the information processing device 11.

As a result, without performing data communications between plural information processing devices, and without adding new commands to the operating system of the information processing device 11, data

stored in a storage device at a primary site can be stored as a backup in a storage device at a remote site. Also, in accordance with the present embodiment, a storage device at a remote site transmits a read request to a storage device at a primary site to thereby perform a copy forming
5 processing. By this, the processing load on the storage device at the primary site during the copy forming processing is alleviated. In other words, in a method in which a storage device at a primary site writes data in a storage device at a remote site, the storage device at the primary site needs to write the data in the storage device at the remote site after it
10 confirms that the storage device at the remote site is ready for forming a pair. For this reason, the processing load on the storage device at the primary site becomes heavier, which would affect the overall performance of the primary site that is performing other primary processings. In contrast, in accordance with the present embodiment, since the storage
15 device at the primary site only has to send data in response to a read request from the storage device at the remote site, the processing load at the storage device at the primary site can be alleviated.

[Restoration]

20 Even after the copy forming processing is performed, the first storage device 10 accepts write requests from the information processing device 11, and updates the data in the primary volumes. For this reason, the data in the primary volumes becomes inconsistent with the data in the auxiliary volumes. As described above, the primary journal stores journal

data for executions performed even after the copy forming processing took place. In this respect, the second storage device 20 copies data stored in the primary journal into the auxiliary journal, and writes the data stored in the auxiliary journal into the auxiliary volumes, such that updates of
5 the data on the primary volumes can be likewise performed on the auxiliary volumes.

Here, a processing to copy data stored in the primary journal into the auxiliary journal by the second storage device 20 is referred to as a “journal acquisition” processing, and a processing to write journal data
10 stored in the auxiliary journal into the auxiliary volume is referred to as a “restoration” processing.

FIG. 13 shows a flowchart of the journal acquisition processing. The information processing device 11 transmits a journal acquisition instruction to the second storage device 20 (S1301). Upon receiving the
15 journal acquisition instruction, the journal acquisition section 708 of the second storage device 20 refers to the pair management table 1001, and obtains a primary journal LUN of the corresponding pair. The journal acquisition section 708 of the second storage device 20 transmits to the first storage device 10 a read request to read the primary journal. Upon
20 receiving a copy of data of the primary journal, the journal acquisition section 708 of the second storage device 20 writes the data in the auxiliary journal (S1302).

Next, referring to FIG. 14, a processing flow of a restoration processing will be described. The information processing device 11

transmits to the second storage device 20 a restore instruction to restore data in the auxiliary journal onto the auxiliary volumes (S1401). Upon receiving the restore instruction, the restore section 706 of the second storage device 20 writes journal data stored in the auxiliary journal into
5 the auxiliary volumes(S1402).

FIG. 15 shows journal data regions 1202 for the primary journal and the auxiliary journal. The journal data regions for the primary journal and the auxiliary journal are defined by the same head LBA and end LBA, respectively. The journal data region 1202 for the primary
10 journal is composed of journal storage completed regions 1502, 1503 and 1504 which store journal data, and purge completed regions 1501 which do not store journal data.

The journal data region 1202 of the auxiliary journal is composed of restoration completed regions 1521 that store journal data that have
15 already been used for restoration in the auxiliary volumes, restore in-progress region 1522 that stores journal data that are designated for restoration, read completed region 1523 that stores journal data that are not designated for restoration, and read in-progress region 1524 that stores journal data that are being read from the primary journal in
20 response to a journal acquisition instruction.

Each of the storage devices 10 and 20 stores journal data in the journal data region 1202 from the head LBA to the end LBA in a chronological order as the journal data is created. When the journal data reaches the end LBA, each of the storage devices 10 and 20 returns to the

head LBA again, and stores journal data from there. In other words, the storage devices 10 and 20 use the journal data regions cyclically between the head LBA and the end LBA.

The first storage device 10 that is equipped with the primary
 5 journal stores a journal-out LBA 1511 which is a head LBA of the journal storage completed regions 1502, 1503 and 1504, and a journal-in LBA 1512 which is a head LBA of the purge completed region 1501. When the journal-out LBA and the journal-in LBA are equal to each other, it means that journal data is not stored in the primary journal.

10 The second storage device 20 that is equipped with the auxiliary journal stores a restoration completed LBA 1531 which is the highest LBA of the restoration completed region 1521, a to-be restored LBA 1532 which is the highest LBA of the restore in-progress region 1522, a read completed LBA 1533 which is the highest LBA of the read completed
 15 region 1523, and a to-be read LBA 1534 which is the highest LBA of the read in-progress region 1524.

In other words, when the restoration completed LBA 1531 and the to-be restored LBA 1532 are equal to each other, it means that a restoration processing instructed by the information processing device 11
 20 has been completed. Also, when the read completed LBA 1533 and the to-be read LBA 1534 are equal to each other, it means that a journal acquisition processing instructed by the information processing device 10 has been completed.

The information processing device 11 can transmit to the first

storage device 10 and the second storage device 20 a request to obtain the processing state of journal. Each of the storage devices 10 and 20 confirms the states of LBAs that indicate the boundaries of the regions described above, and responds to the request.

5 Also, since the storage devices 10 and 20 use the journal data regions cyclically as described above, regions that become unnecessary need to be released. The processing to release a region is called a “purge” processing. Each of the storage devices 10 and 20 can perform a purge processing by changing addresses of LBAs that indicate the boundaries of
10 the regions. The first storage device 10 can purge the journal storage completed region 1502, among the journal storage completed regions 1502, 1503 and 1504 of the primary journal, which the second storage device 20 has completed acquiring the journal data into the auxiliary journal. In this case, the first storage device 10 changes the journal-out LBA 1511 to
15 the head LBA of the journal storage completed region 1503, such that the journal storage completed region 1502 becomes the purge completed region 1501. The second storage device 20 treats the restoration completed region 1521 of the auxiliary journal as a region that is purged, and stores the journal data obtained in response to the journal acquisition
20 instruction in the restoration completed region 1521.

Referring to a flowchart in FIG. 16, flows of the journal acquisition processing and the restoration processing will be described. The information processing device 11 transmits to the first storage device 10 a request to obtain the processing status of the primary journal (S1601).

The information processing device 11 transmits to the first storage device 10 a read request to read edited data of a command device at which the processing status of the primary journal is set (S1602). Upon receiving the edited data of the command device from the first storage device 10 (S1603),
5 the information processing device 11 transmits to the second storage device 20 a journal acquisition request to obtain journal data starting at the journal-out LBA 1511 to a LBA immediately before the journal-in LBA 1512 (S1604). The information processing device 11 transmits to the second storage device 20 a request to obtain the processing status of the
10 auxiliary journal (S1605). The information processing device 11 transmits to the second storage device 20 a read request to read edited data of a command device at which the processing status of the auxiliary journal is set (S1606). Upon receiving the edited data of the command device from the second storage device 20 (S1607), the information processing device 11
15 compares the read completed LBA 1533 and the to-be read LBA 1534 set in the edited data, to confirm whether or not acquisition of the journal data has been completed (S1608). When the acquisition of the journal data has been completed, the information processing device 11 transmits to the second storage device 20 a restore request to restore journal data up
20 to the read completed LBA 1533 (S1609). Then, the information processing device 11 transmits to the first storage device 10 a purge request to purge the journal data up to the read completed LBA 1533 (S1610). The information processing device 11 repeats the journal acquisition processing and restoration processing.

By the processings described above, updated data in a storage device at a primary site can be reflected on a storage device at a remote site without performing data communications between multiple information processing devices, and without adding new commands to the operating system of the information processing devices. It is noted that, with an instruction from the information processing device 11 that is communicatively connected to a storage device at a remote site, the storage device at the remote site can obtain journal data from a storage device at a primary site and restore the data.

10

[Swap]

Let us assume that a primary volume of the first storage device 10 and an auxiliary volume of the second storage device 20 form a pair by an instruction from an information processing device 11 (hereafter referred to as a “first information processing device”) that is communicatively connected to the first storage device 10. In this case, if a failure occurs in the first information processing device 10, an information processing device 11 (hereafter referred to as a “second information processing device”) that is communicatively connected to the second storage device 20 can continue processings that have been performed by the first information processing device, using the auxiliary volume of the pair. In this instance, the second information processing device switches the relation between the primary volume and the auxiliary volume. In other words, a pair is formed with the logical volume 209 of the second storage

15

20

device 20 being a primary volume and the logical volume 209 of the first storage device 10 being an auxiliary volume. Such a processing to switch the pair relation is called a “swap” processing.

Referring to FIGS. 17 and 18, a flow of processings to swap a pair
5 will be described. As shown in FIG. 17, the second information processing device 11 transmits a pair swap instruction to the first storage device 10 and the second storage device 20 (S1701, S1702). Upon receiving the pair swap instruction, the journal stop section of the first storage device 10 stops storing journals in the primary volume. Also, the pair management
10 section 704 of the first storage device 10 swaps the primary volume and the auxiliary volume registered in the pair management table 1001. Similarly, the pair management section 704 of the second storage device 20 swaps the primary volume and the auxiliary volume registered in the pair management table 1001. The journal storage section 707 of the
15 second storage device 20 starts storing journals of the logical volume 209 of the second storage device 20, which defines the primary volume.

FIG. 18 shows a state in which a pair is formed with the logical volume 209 of the second storage device 20 being the primary volume and the logical volume 209 of the first storage device 10 being the auxiliary
20 volume, as a result of the swap processing performed by the first storage device 10 and the second storage device 20 which received the pair swap instruction.

The swap processing performed by the second information processing device 20 and the storage devices 10 and 20 will be described in

detail with reference to flowcharts in FIGS. 19 through 21. The second information processing device executes the journal acquisition processing and the restoration processing described above (S1901). The second information processing device transmits to the second storage device 20 a request to obtain the processing status of the auxiliary journal (S1902).
5 The second information processing device transmits to the second storage device 20 a read request to read edited data of a command device in which the processing status of the auxiliary journal is set (S1903). Upon receiving the edited data of the command device from the second storage device 20 (S1904), the second information processing device compares the restoration completed LBA 1531 and the to-be restored LBA 1532 set in the edited data, to confirm if the restoration processing has been completed (S1905). If the restoration has been completed, the second information processing device transmits a pair swap request to the first storage device 10 and the second storage device 20 (S1906, S1907).
15 Upon receiving the pair swap request, the first storage device 10 stops its journal acquisition processing with respect to the primary volume (S2001), and swaps the relation between the copy source and the copy destination registered in the pair management table 1001 (S2002). Also, the second storage device 20, that has received the pair swap request, swaps the relation between the copy source and the copy destination registered in the pair management table 1001 (S2101), and starts a journal acquisition processing with respect to the primary volume of the second storage device 20 (S2102).

Let us consider as an example an information processing system that is composed of a primary site and a remote site. The primary site is equipped with a first information processing device and a first storage device 10, and the remote site is equipped with a second information processing device and a second storage device 20. When a failure occurs in the first information processing device, the second information processing device uses the second storage device 20 to continue primary processings performed at the primary site. The second information processing device may instruct the first storage device 10 and the second storage device 20 to execute the swap instruction described above, such that the second storage device 20 is used for the primary processings, and data on the second storage device 20 can be stored as a backup in the first storage device 10. Furthermore, since the data on the second storage device 20 is stored as a backup in the first storage device 10, the execution of the primary processings can be quickly switched to the primary site, when the first information processing device is recovered from the failure.

Also, since the pair swap instructions from the information processing device 11 to the storage devices 10 and 20 are provided using read/write commands with which the information processing device 11 is equipped, there is no need to add new commands to the operating system on the information processing device 11.

[Three-site Structure with Command Device System]

The storage device system described above transfers commands

between two storage devices using virtual volumes and command devices. Furthermore, a virtual volume structure can be set up among three or more storage devices so that commands can be transferred among three or more storage devices.

5 FIG. 22 shows a storage device system in accordance with an embodiment of the present invention, which includes, for example, a first storage device 10, a second storage device 20 and a third storage device 25.

 A logical volume 2201 of the first storage device 10 is a virtual volume of a logical volume 2202 of the second storage device 20, the logical
10 volume 2202 of the second storage device 20 is a virtual volume of a logical volume 2203 of the third storage device 25. A third logical volume 2203 is a command device. By connecting the virtual volumes in this manner, commands can be transferred from an information processing device 11 to the third storage device 25.

15 Accordingly, by using the single information processing device 11 and three storage devices 10, 20 and 25, a storage device system that can handle disaster recovery can be structured. FIG. 23 indicates the storage device system.

 The first storage device 10, the second storage device 20 and the
20 third storage device 25 are equipped with logical volumes 2301, 2302 and 2303, respectively. In this example, the logical volume 2301 of the first storage device 10 and the logical volume 2302 of the second storage device 20 form a synchronous pair with the logical volume 2301 of the first storage device as being a primary logical volume.

The following is a description of the synchronous pair. FIG. 24 indicates a synchronous pair management table 2401. The synchronous pair management table 2401 is equipped with entries of "COPY SOURCE DEVICE," "COPY DESTINATION DEVICE," "PRIMARY LUN," and
5 "AUXILIARY LUN." When the logical volume 2301 of the first storage device 10 and the logical volume 2302 of the second storage device 20 form a synchronous pair, the first storage device 10 is set as a copy source device, the second storage device 20 as a copy destination device, a LUN of the logical volume 2301 of the first storage device 10 as a primary LUN,
10 and a LUN of the logical volume 2302 of the second storage device 20 as an auxiliary LUN.

Upon receiving a data write request from the information processing device 11 to write data in the logical volume 2301 of the first storage device 10, which is a primary volume, the first storage device 10
15 refers to the synchronous pair management table 2401, and sends the data write request to the second storage device 20 to write the data in the logical volume 2302 of the second storage device 20, which is an auxiliary volume corresponding to the primary volume. Upon receiving a data write completion notification for the data write request from the second storage
20 device 20, the first storage device 10 sends the data write completion notification to the information processing device 11. In other words, when the write completion notification is sent to the information processing device 11, the same data have been written in the primary volume and the auxiliary volume. Accordingly, by using the synchronous pair, data

can be backed up in the second storage device 20, without losing the data of the first storage device 10. However, in making a synchronous pair, if there is a great distance between the first storage device 10 and the second storage device 20, transmission of a write completion notification to the information processing device 11 may be delayed, and the processing by the information processing device 11 may be affected.

Accordingly, a synchronous pair is formed when the first storage device 10 and the second storage device 22 are located in a short distance from each other, and a pair using the journals described above (hereafter referred to as an “asynchronous pair”) is formed with the second storage device 20 and the third storage device 25. The information processing device 11 sends to the second storage device 20 and the third storage device 25 a command to form a pair with the logical volume of the second storage device 20 being as a primary volume and the logical volume of the third storage device 25 as being an auxiliary volume, using the virtual volumes and command devices. After the pair is formed, the information processing device 11 sends a command to obtain and restore a journal in the pair to the third storage device 25, by using the virtual volumes and the command devices. Accordingly, data of the first storage device 10 can be backed up in the third storage device 25 that may be installed at a great distance from the first storage device 10. Also, neither the third storage device 25 nor the second storage device 20 that is located intermediate between the first and third storage devices 10 and 25 requires an information processing device. It is noted that an information

processing device for backup may be connected to the third storage device 25, and processings may be continued using data in the third storage device 25 when the first storage device 10 fails.

A data backup method in which commands are transferred among storage devices by using virtual volumes and command devices is
5 described above. In the description above, the information processing device 11 sends commands to the storage devices 10, 20 and 25. However, each of the storage devices 10, 20 and 25 may be equipped with a command setting section 701 and a command transmission section 702.
10 For example, in the structure indicated in FIG. 23, the first storage device 10 can send a command to be executed at the third storage device 25 to the command device of the third storage device 25 via the virtual volume of the second storage device 20. By this, for example, the first storage device 10 can send to the third storage device 25 a command to confirm the
15 status of the third storage device 25, such that the first storage device 10 can detect the status, such as, a failure of the third storage device 25. If a failure occurs in the third storage device 25, the first storage device 10 can send to the second storage device 20 and the third storage device 25 a command to cancel the above-described asynchronous pair between the
20 second storage device 20 and the third storage device 25.

[Method for Designating Transfer Destination Address]

Next, a method to designate an address of a storage device of a transfer destination at the time of transferring a command. FIG. 25

indicates a path information management table 2501 that is stored in the information processing device 11. The path information management table 2501 includes columns of "DEVICE" and "PATH." In the "DEVICE" column, addresses of the storage devices 10 and 20 are stored. For
5 example, an address may be information that is composed of a product serial number, a port number and the like of each of the storage devices 10 and 20. The "PATH" column stores information indicating as to how each of the storage devices 10 and 20 is connected to the information processing device. For example, according to the path information management table
10 2501 shown in FIG. 25, the path to the first storage device 10 is indicated as being "DIRECT" and the path to the second storage device 20 is indicated as being "FIRST." This means that the information processing device 11 is directly connected to the first storage device 10, and is connected to the second storage device 20 through the first storage device
15 10.

FIG. 26 shows a command interface 2601 which is a data format used when transferring a command from the information processing device 11 to the storage devices 10 and 20. The command interface 2601 is composed of an identification number, a transfer destination parameter, a
20 control parameter and an input parameter. The identification number is a number for identifying the data, which is appended by the information processing device 11 when generating the data. Addresses of the storage devices 10 and 20 to which the data is sent are set at the transfer destination parameter. The control parameter and the input parameter

set parameters similar to those of the command device interface 601 that is used in the command device.

An outline of processings to be executed when the information processing device 11 transfers a command to the second storage device 20 will be described with reference to FIG. 27. The information processing device 11 is equipped with a data generation section 2701 and a data transfer section 2702. Each of the storage devices 10 and 20 is equipped with a command analysis section 2703, a data transfer section 2704 and a command execution section 2705.

The data generation section 2701 of the information processing device 11 refers to the path information management table 2501, refers to the path to the second storage device 20, and recognizes that the second storage device 20 is connected via the first storage device 10. The data generation section 2701 generates data 2710 in the command interface 2601 format in which the address of the second storage device 20 is set as a transfer destination address and a first command is set at a process number. The data generation section 2701 notifies the data transfer section 2702 to transfer the data 2710 to the first storage device 10. The data transfer section 2702 transfers the data 2710 to the first storage device 10.

The command analysis section 2703 of the first storage device 10, upon receiving the data 2710, notifies the data transfer section 2704 of the first storage device 10 to transfer the data 2710 to the second storage device 20 that is set as the transfer destination address. The data transfer

section 2704 of the first storage device 10 generates data 2711 by deleting the transfer destination parameter from the data 2710, and transfers the data 2711 to the second storage device 20. The command analysis section 2703 of the second storage device 20, upon receiving the data 2711, obtains
 5 the first command from the command parameter, as the data 2711 does not contain a transfer destination parameter. The command analysis section 2703 of the second storage device 20 notifies the command execution section 2705 of the second storage device 20 to execute the first command. The command execution section 2705 of the second storage
 10 device 20, upon receiving the notification from the command analysis section 2703, executes the first command.

When the first command causes an output result, the command execution section 2705 of the second storage device 20 stores the output result as edited data 2712. In this example, the same identification
 15 number is appended to all of the data 2710, 2711 and 2712. When the first command causes an output result, the information processing device 11 can obtain the output result of the first command by designating an identification number at the first storage device 10 and sending a send command to send the edited data 2712.

20

[Method for Designating Transfer Destination Address in Three-site Structure]

The method for transferring a command between two storage devices by designating a transfer destination address has been described

above. Further, a command can be transferred among three or more storage devices by designating multiple transfer destination addresses.

FIG. 28 shows a flow of processings to transfer a command among three storage devices, i.e., the first storage device 10, the second storage device 20 and the third storage device 25. The information processing device 11 is communicatively connected to the first storage device 10, the first storage device 10 is communicatively connected to the second storage device 20, and the second storage device 20 is communicatively connected to the third storage device 25. In this example, the path information management table 2501 of the information processing device 11 sets paths indicated in FIG. 29.

A process flow in which the information processing device 11 transfers a first command to the third storage device 25 is described below. The data generation section 2701 of the information processing device 11 refers to the path information management table 2501 and recognizes that the third storage device 25 is connected via the first storage device 10 and the second storage device 20. The data generation section 2701 generates data 2801 in the command interface 2601 format in which the second storage device 20 and the third storage device 25 are set as transfer destination addresses, and a first command is set as a process number. The data generation section 2701 notifies the data transfer section 2702 to transfer the generated data to the first storage device 10 that is at the head address in the transfer path. The data transfer section 2702, upon receiving the notification, sends the data 2801 to the storage device 10.

The command analysis section 2703 of the first storage device 10, upon receiving the data 2801, obtains the address of the second storage device 20 which is set at the head of the transfer destination parameter, and notifies the data transfer section 2704 of the first storage device 10 to
5 send the data to the second storage device 20. The data transfer section 2704, upon receiving the notification, generates data 2808 by deleting the transfer destination address of the second storage device 20 from the data 2801, and sends the data 2802 to the second storage device 20.

The second storage device 20 obtains the address of the third
10 storage device 25 which is set at the head of the transfer destination parameter just as does the first storage device 10, generates data 2803 by deleting the third storage device 25 from the transfer destination parameter, and sends the data 2803 to the third storage device 25.

The command analysis section 2703 of the third storage device 25,
15 upon receiving the data 2803, obtains the process number of the first command from the control parameter as the data 2803 does not contain a transfer destination address, and notifies the command execution section 2705 of the third storage device 25 to execute the first command. The command execution section 2705 of the third storage device 25, upon
20 receiving the notification, executes the first command.

In this manner, by setting multiple transfer destination addresses in the transfer destination parameter, a command can be transferred among three or more storage devices.

[Processings of Method for Designating Transfer Destination Address]

FIG. 30 shows a flowchart of processings executed by the information processing device 11 when a command is transferred with a transfer destination address being designated.

5 First, the information processing device 11 generates an identification number and sets the identification number in a command interface 2601 (S3001). The identification number can be any number that can uniquely identify the command interface 2601 generated. The information processing device 11 refers to the path information
10 management table 2501, and obtains a path for a storage device to which a command is to be transferred (S3002). The information processing device 11 sets one or multiple transfer destination addresses at the transfer destination parameter based on the path information obtained (S3003).
Next, the information processing device 11 sets a process number of the
15 command to be executed by the storage device at the transfer destination and a presence/absence of edited data of the command (S3004). Then, the information processing device 11 sends the data of the command interface 2601 to a storage device at the head address in the transfer path (S3005).
The information processing device 11 confirms whether the command
20 results in edited data (S3006), and ends the processing when no edited data exists. When edited data exists, the information processing device 11 designates the identification number and sends an edited data send request to the storage device at the head address in the transfer path (S3007). The information processing device 11 waits until it receives the

edited data (S3008), and ends the processing upon receiving the edited data.

FIG. 31 shows a flowchart of processings performed at the storage devices 10, 20 and 25, when a command is transferred with a transfer
5 destination address being designated.

Each of the storage devices 10, 20 and 25 may perform the following processings. Upon receiving the data of the command interface (S3101), each storage device stores the data in a cache memory (S3102). The storage device confirms whether the data contains a transfer destination
10 parameter (S3103). If the transfer destination parameter exists, the storage device obtains a transfer destination address at the head of the transfer destination parameter (S3104). Next, the storage device deletes the transfer destination address at the head of the transfer destination parameter from the data (S3105), and sends the data to the transfer
15 destination address previously obtained (S3106). The storage device confirms whether the command results in edited data (S3107). If edited data exists, the storage device executes a processing to obtain the edited data (S3108), deletes the data from the cache memory (S3109), and ends the processing. If no edited data exists, the storage device does not
20 perform the processing to obtain edited data, deletes the data from the cache memory (S3109), and ends the processing.

When a transfer destination parameter is not set in the received command interface 2601, the storage device (which may be 10, 20 or 25) obtains a process number of the control parameter (S3110), and executes a

command designated by the process number (S3111). The storage device (10, 20, 25) confirms whether the command results in edited data (S3112). If edited data exists, the storage device (10, 20, 25) generates edited data set with the identification number set in the command interface 2601 (S3113), executes a process to obtain edited data (S3108), deletes the data from the cache memory (S3109), and ends the processing. If edited data does not exist, the storage device (10, 20, 25) deletes the data from the cache memory (S3109) and ends the processing.

FIG. 32 shows a flowchart of the processing to obtain edited data performed by the storage device (10, 20, 25).

In the process to obtain edited data, the storage device (10, 20, 25) waits until it receives a edited data obtaining request to obtain edited data (S3201). Upon receiving the request to obtain edited data, the storage device refers to the command interface 2601 at the identification number that is sent in the edited data obtaining request (S3202). The storage device (10, 20, 25) confirms if the command interface 2601 referred to contains a transfer destination parameter (S3203). If the transfer destination parameter exists, it is not the storage device (10, 20, 25) that executed the command, such that the storage device (10, 20, 25) obtains a transfer destination address at the head of the transfer destination parameter (S3204), and sends the edited data obtaining request to the obtained transfer destination address (S3205). The storage device (10, 20, 25) waits until it receives edited data (S3206), and, when it receives the edited data, the storage device sends the edited data received to the

transmission source of the edited data obtaining request (S3207). When
 no transfer destination parameter exists, it is the storage device that
 executed the command, the storage device (10, 20, 25) obtains edited data
 of the identification number (S3208), and sends obtained edited data to
 5 the transmission source of the edited data obtaining request (S3209). By
 this, the edited data is sent to the information processing device 11 back
 along the path in which the command is transferred.

In the processing indicated in FIG. 31, when the command interface
 2601 is transferred, the transfer destination address at the head thereof is
 10 deleted. But, the data of the command interface 2601 can be sent to the
 transfer destination as is without deleting the transfer destination
 address. FIG. 33 and FIG. 34 show flowcharts of processings performed in
 this situation by the storage device (10, 20, 25).

FIG. 33 indicates processings performed by the storage device (10,
 15 20, 25) as it receives the command interface 2601, and its basic process
 flow is generally the same as the process flow indicated in FIG. 31. The
 process flow indicated in FIG. 33 differs from the process flow in FIG. 31
 in that the storage device (10, 20, 25) that has received the data of the
 command interface 2601 needs to judge as to whether the storage device
 20 itself is in a transfer path or whether the storage device should execute a
 command.

More specifically, the storage device (10, 20, 25) confirms whether
 the last transfer destination address is the address of its own (S3303), and
 determines that the storage device (10, 20, 25) is in a transfer path if the

last transfer destination address is not its own address, and determines that the storage device (10, 20, 25) should execute a command if the last transfer destination address is its own address. When the storage device (10, 20, 25) is in the transfer path, the storage device confirms whether the transfer destination address includes its own address (S3304); if its own address is included, the storage device obtains a transfer destination address next to the address of its own (S3305); and if its own address is not included, the storage device obtains the head transfer destination address (S3306). Then, the storage device (10, 20, 25) sends the data of the command interface 2601 to the obtained transfer destination address (S3307). When the storage device (10, 20, 25) is to execute a command, the storage device (10, 20, 25) obtains a process number of the control parameter (S3311), and executes a command designated by the process number (S3312). Other processings are basically the same as those of the processings indicated in FIG. 31 except an edited data obtaining process (S3309).

FIG. 34 shows a flowchart of the edited data obtaining process performed when the transfer destination address is not deleted. The edited data obtaining process in FIG. 34 differs from the process indicated in FIG. 31 in that the storage device that has received an edited data obtaining request needs to judge whether the storage device (10, 20, 25) itself is in a transfer path or whether the storage device (10, 20, 25) itself has executed the command. The storage device (10, 20, 25) judges by the same method indicated in FIG. 33 (S3403); if the storage device (10, 20,

25) is in a transfer path, the storage device (10, 20, 25) sends the edited data obtaining request to the next transfer destination (S3405 – S3408); and if the storage device (10, 20, 25) has executed the command, the storage device (10, 20, 25) obtains edited data of an identification number
5 set in the edited data obtaining request (S3411). The other processings are the same as those indicated in FIG. 32.

[Determination of Shortest Transfer Path]

Next, a description will be made as to a process in which the storage
10 device (10, 20, 25) determines the shortest transfer path and sends a command interface 2601 in the transfer path determined.

FIG. 35 shows a flowchart of processings performed when the first storage device 10 receives data 3501 of a command interface from the information processing device 11. The shared memory 204 of the first
15 storage device 10 stores a connection information management table 3601 indicated in FIG. 36. The connection information management table 3601 can be set by a system administrator through the management terminal 207. The connection information management table 3601 stores addresses of the storage devices 10 and 20 that are connected. For example, the
20 connection information management table 3601 in FIG. 36 indicates that the first storage device 10 is connected to the second storage device 20 and the third storage device 25.

The first storage device 10, upon receiving the data 3501 of the command interface from the information processing device 11, refers to

the connection information management table 3601, and obtains the address of the third storage device 25 which is set at the last of the transfer destination parameter. The first storage device 10 judges that data can be directly sent to the third storage device 25 because the
 5 address of the third storage device 25 is stored in the connection information management table 3601. The first storage device 10 generates data 3502 by deleting the transfer destination addresses of the third storage device 25 and the second storage device 20, and directly sends the data 3502 to the third storage device 25 without passing it
 10 through the second storage device 20.

FIG. 37 shows a flowchart of processings in which the storage device (which may be 10, 20 or 25) determines the shortest path and transfers data of a command interface, which differs from the processing indicated in FIG. 31 only in that the storage device (10, 20, 25) judges the
 15 shortest transfer path.

The storage device (10, 20, 25) obtains the number of transfer destination addresses set in the transfer destination parameter, and sets the obtained number as a variable "N" (S3704). Then, the storage device (10, 20, 25) obtains the N-th transfer destination address (S3705), and
 20 confirms whether the obtained transfer destination address is stored in the connection information management table 3601 (S3706). When the obtained transfer destination address is not stored in the connection information management table 3601, the number N is reduced by 1 (S3707), and the storage device (10, 20, 25) repeats the processings to

obtain the N-th transfer destination address and to confirm whether the
obtained transfer destination address is stored in the connection
information management table 3601 (S3705, S3706). As a result, the
storage device (10, 20, 25) can obtain the shortest transfer path. Then, the
5 storage device (10, 20, 25) deletes the transfer destination addresses
before the N-th transfer destination address from the data of the command
interface 2601 (S3708), and sends the data to the N-th transfer destination
address. The other processings are the same as those indicated in FIG. 31.
Also, the edited data obtaining process (S3711) is the same as the process
10 in FIG. 32.

In this manner, the storage device (10, 20, 25) determines the
shortest transfer path, and transfers a command along the shorted
transfer path, such that the data transfer amount among the storage
devices can be reduced, and the transfer time for transferring commands
15 can be shorted.

[Execution of Command at Each of the Storage Devices]

Next, a description will be made as to a method in which the
information processing device 11 sends commands to a plurality of storage
20 devices 10, 20 and 30 with a single instruction.

FIG. 38 indicates a command interface 3801 for sending commands
to a plurality of storage devices 10, 20 and 25. The command interface
3801 has a structure in which a plurality of command interfaces 2601
shown in FIG. 26 are connected.

FIG. 39 shows an example of processings in which the information processing device 11 sends a first command to the second storage device 20 and a second command to the third storage device 25. The information processing device 11 sets the second storage device 20 at the transfer destination address of the first transfer destination parameter, and the first command at the process number of the first control parameter. Then, the information processing device 11 sets the third storage device 25 at the transfer destination address of the second transfer destination parameter, and the second command at the process number of the second control parameter. The information processing device 11 sends data 3901 thus generated to the first storage device 10 that is located at the head of the transfer path.

The first storage device 10, upon receiving the data 3901, obtains the transfer destination address of the first transfer destination parameter, and sends data 3902 by deleting the head transfer destination parameter from the data 3901 to the second storage device 20. The second storage device 20, upon receiving the data 3902, executes the first command that is set in the first control parameter. Then the second storage device 20 deletes the head control parameter in the data 3902, and obtains the transfer destination address of the transfer destination parameter. The second storage device 20 sends to the third storage device 25 data 3903 by deleting the head transfer destination parameter from the data 3902. The third storage device 25, upon receiving the data 3903, executes the second command set in the head control parameter.

FIG. 40 shows a flowchart of processings performed by the first information processing device 11 when sending commands to a plurality of storage devices 10, 20 and 25. The information processing device 11 refers to the path information management table 2501, and obtains paths to the storage devices 10, 20 and 25 to which the commands are sent (S4001).
 5 Then, the information processing device 11 repeats executions to set a transfer destination address in the transfer destination parameter (S4002) and to set a command in the control parameter (S4003) in the order from the head storage device 10 to the storage devices 20 and 25 in the path until there is no transfer destination remained (S4004). The information
 10 processing device 11 sends the data of the command interface 3801 thus generated to the head storage device 10 and the storage devices 20 and 25 in the transfer path (S4005).

FIG. 41 shows a flowchart of processings performed at each of the storage devices 10, 20 and 25 when commands are sent to the plurality of
 15 storage devices 10, 20 and 25. The storage device (which may be 10, 20 or 25), upon receiving data of the command interface 3801 (S4101), stores the data in the cache memory (S4102). The storage device (10, 20, 25) confirms whether the head of the data is a transfer destination parameter (S4103). If the head of the data is a transfer destination parameter, the
 20 storage device (10, 20, 25) obtains a transfer destination address of the head transfer destination parameter (S4104), deletes the head transfer destination parameter from the data (S4105), and sends the data to the obtained transfer destination address (S4106). When the head of the data

is not a transfer destination parameter, the storage device (10, 20, 25) obtains a process number of the head control parameter (S4107), and executes a command designated by the process number (S4108). The storage device (10, 20, 25) deletes the head control parameter (S4109), and
5 confirms whether a further transfer destination parameter exists (S4110). When the further transfer destination parameter exists, the storage device (10, 20, 25) performs processings from obtaining a transfer destination address to sending data to the transfer destination address (S4104 – S4106).

10 In this manner, by sending commands to a plurality of storage devices 10, 20 and 25, for example, in the case of forming asynchronous pairs among the storage devices, the information processing device 11 can perform controls, such as, simultaneously sending pair forming commands to the storage devices 10, 20 and 25 that form primary volumes and to the
15 storage devices 10, 20 and 25 that form auxiliary volumes in asynchronous pairs. In other words, pairs can be readily formed without communicating among plural information processing devices.

In the above described embodiment, commands are transferred through designating transfer destination addresses. In the embodiment
20 described above, the information processing device 11 sends commands to the storage devices 10, 20 and 25. However, each of the storage devices 10, 20 and 25 may be equipped with a data generation section 2701 and a data transmission section 2702.

By using the method of transferring commands through designating

transfer destination addresses in a manner described above, commands can be executed by each of the storage devices 10, 20 and 25 which may not be connected to an information processing device, just as does the storage device system transfer commands through using virtual volumes and command devices. By this, data backup and pair operations can be conducted without communicating data among multiple information processing devices. Also, when commands are transferred through designating transfer destination addresses, dedicated logical volumes such as command devices do not need to be provided, and therefore data areas to be allocated to users would not be reduced. Also, in the case of command devices, there is a possibility that the performance of the storage devices 10, 20 and 25 may lower because data input/output requests may concentrate on specified logical volumes. In contrast, when transferring commands through designating transfer destination addresses, cache memories are used, such that a reduction in the performance of the storage devices 10, 20 and 25 can be prevented.

While the description above refers to particular embodiments of the present invention, it will be understood that these embodiments are presented for ready understanding of the present invention and many modifications may be made without departing from the spirit thereof. The accompanying claims are intended to cover such modifications as would fall within the true scope and spirit of the present invention. For example, in the present embodiments, commands that are transferred are commands relating to pairs of logical volumes. However, commands to be

transferred are not limited to commands relating to pairs, but may be any commands that are executable by the storage devices 10, 20 and 25.

The presently disclosed embodiments are therefore to be considered in all respects as illustrative and not restrictive, the scope of the invention
5 being indicated by the appended claims, rather than the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.